Logical reasoning and programming, task I

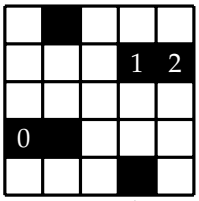
(October 8, 2018)

Problem

Your task is to produce a solver for a puzzle called Light Up (or Akari) using a SAT solver.

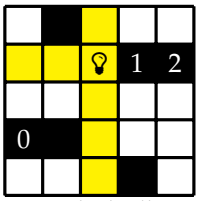
You have a grid, in our case always $n \times n$, with white and black cells. The goal is to place light bulbs in white cells in such a way that all white cells are lit. A light bulb illuminates all white cells visible horizontally and vertically from it and all black cells block light. Moreover, no light bulb is allowed to illuminate another light bulb (but a cell can be illuminated by more light bulbs). Black cells (walls) can contain numbers (0, 1, 2, 3, or 4) that say exactly how many light bulbs have to be placed to its neighbors (horizontal and vertical, diagonal neighbors do not count).

For example, if we have the following assignment



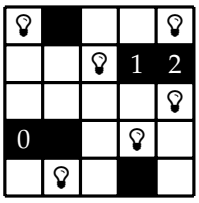
Example

then we can place a light bulb as follows and it illuminates the yellow cells



Light bulb

In our example, the following is the only possible solution.



Solution

For simplicity, you can assume that there is at most one solution; it is possible that no solution exists.

Program

You should upload an archive to BRUTE that contains an executable script `lightup` that expects an input string on `stdin` and produces a solution to `stdout`.

It is expected that you use Python (use `python2` or `python3`), but MATLAB 9.2 (use `matlab`) should also work. You can use

- PycoSAT in Python, `import pycosat`,
- MiniSat, command `minisat`,
- PicoSAT, command `picosat`,

as solvers. You are allowed to use other solvers included in your archive.

Every input has a maximal time for which you can solve it, however, the given time should be more than enough for solving the problem using a decent SAT solver with a non-optimized encoding.

Non-standard settings can be discussed individually.

Input

An input is a string of length $n \times n$. In our example it is

```
WWWBW0BWWWWWWWWWW12WBWWW
```

where `W` is a white cell, `B` is a black cell without a number, and `0`...`4` are black cells with a given number in them. A cell (x, y) is described by a character at the position $(n \cdot y) + x$ in the string, we start counting from zero.

Output

The output of your solver is the string you receive on the input where your placement of light bulbs is indicated by replacing corresponding `W`s by `L`s.

Hence you are supposed to produce

```
WLWBW0BWLWWWWWLWWL12LBWWL
```

If no solution is possible, then just produce string

```
0
```